Approximating the Metric TSP in Linear Time

Davide Bilò[1], Luca Forlizzi[2], and Guido Proietti[2,3]

[1] Institut für Theoretische Informatik, ETH, Zürich, Switzerland
[2] Dipartimento di Informatica, Università di L'Aquila, Italy
[3] Istituto di Analisi dei Sistemi ed Informatica, CNR, Roma, Italy
{davide.bilo,forlizzi,proietti}@di.univaq.it

Abstract. Given a metric graph $G = (V, E)$ of n vertices, i.e., a complete graph with an edge cost function $c : V \times V \rightarrow \mathbb{R}_{\geq 0}$ satisfying the triangle inequality, the *metricity degree* of G is defined as $\beta = \max_{x,y,z \in V} \left\{ \frac{c(x,y)}{c(x,z)+c(y,z)} \right\} \in \left[\frac{1}{2}, 1\right]$. This value is instrumental to establish the approximability of several NP-hard optimization problems definable on G, like for instance the prominent *traveling salesman problem*, which asks for finding a Hamiltonian cycle of G of minimum total cost. In fact, this problem can be approximated quite accurately depending on the metricity degree of G, namely by a ratio of either $\frac{2-\beta}{3(1-\beta)}$ or $\frac{3\beta^2}{3\beta^2-2\beta+1}$, for $\beta < \frac{2}{3}$ or $\beta \geq \frac{2}{3}$, respectively. Nevertheless, these approximation algorithms have $O(n^3)$ and $O(n^{2.5} \log^{1.5} n)$ running time, respectively, and therefore they are superlinear in the $\Theta(n^2)$ input size. Thus, since many real-world problems are modeled by graphs of huge size, their use might turn out to be unfeasible in the practice, and alternative approaches requiring only $O(n^2)$ time are sought. However, with this restriction, all the currently available approaches can only guarantee a 2-approximation ratio for the case $\beta = 1$, which means a $\frac{2\beta^2}{2\beta^2-2\beta+1}$-approximation ratio for general $\beta < 1$. In this paper, we show how to enhance –without affecting the space and time complexity– one of these approaches, namely the classic *double-MST* heuristic, in order to obtain a 2β-approximate solution. This improvement is effective, since we show that the double-MST heuristic has in general a performance ratio strictly larger that 2β, and we further show that *any* re-elaboration of the shortcutting phase therein provided, cannot lead to a performance ratio better than 2β.

Keywords: Traveling Salesman Problem, Metric Graphs, NP-hardness, Linear-time Approximation Algorithms.

1 Introduction

The *Traveling Salesman Problem* (TSP, for short) is one of the most prominent and studied combinatorial optimization problems. It is defined as follows: Given a complete, undirected graph $G = (V, E)$ of n vertices, with an edge cost function $c : V \times V \rightarrow \mathbb{R}_{\geq 0}$, find a Hamiltonian cycle $H = (V, E(H))$ of G (i.e., a simple cycle that spans all the vertices of G) of minimum total *cost* $c(H) = \sum_{e \in E(H)} c(e)$.

H. Broersma et al. (Eds.): WG 2008, LNCS 5344, pp. 43–54, 2008.

The TSP is intractable and does not admit any polynomial-time constant-ratio approximation algorithm, unless $\mathsf{P} = \mathsf{NP}$. On the other hand, when the edge cost function induces a metric on G (i.e., for any $x, y, z \in V, c(x,z) \leq c(x,y) + c(y,z)$), things improve sensibly. More specifically, in such a case the problem is known as the *metric* TSP, and it can be approximated by the Christofides algorithm [4] within an approximation factor of 3/2, but still it remains NP-hard to find an approximate solution for it within a factor better than 220/219 [9]. In some special metric cases, however, the approximability of the problem becomes really effective. For instance, when vertices of G are actually points in a fixed-dimension Euclidean space, and the edge cost function is their Euclidean distance, then the problem (also known as *geometric* TSP) admits a polynomial-time approximation scheme [1].

Getting back to the general metric case, it is interesting to notice that one can define suitable approximation algorithms depending on the *metricity degree* of G, which is defined as $\beta = \max_{x,y,z \in V} \left\{ \frac{c(x,y)}{c(x,z)+c(y,z)} \right\} \in \left[\frac{1}{2}, 1\right]$. More precisely, for $\frac{1}{2} \leq \beta \leq \frac{2}{3}$, there exists a $\frac{2-\beta}{3(1-\beta)}$-approximation algorithm [2],[1] while for $\frac{2}{3} < \beta \leq 1$, the best approximation algorithm is still the Christofides' one, with a ratio of $\frac{3\beta^2}{3\beta^2-2\beta+1}$. In the following, we will therefore assume that the input graph is β-*metric* (i.e., its metricity degree is equal to β), and the metric TSP with this additional parameter will be correspondingly named the β-*metric traveling salesman problem* (β-MTSP for short, also known as *strengthened-metric* TSP).

Both the aforementioned best-known approximation algorithms for the β-MTSP have a shortcoming, however: their time complexity is superlinear in the input size. Indeed, the former runs in $O(n^3)$ time,[2] while the latter runs in $O(n^{2.5} \log^{1.5} n)$ time [8]. From a practical point of view, this might result in a severe drawback, since often the input graph is massive. For instance, the well-known benchmark TSPLIB of TSP instances [10] contains several input graphs with order of 10^5 vertices (most of them coming from roadmaps, and then most likely metric), for which these algorithms become computationally expensive. In such a case, a linear-time approximation algorithm would be desirable, even if a (possibly small) price in terms of performance ratio has to be paid. In this paper, we exactly aim at this goal. We emphasize that insisting on this trade-off is not completely novel, and linear-time constrained approximation algorithms have been already developed in the past, but as far as we know, this was done only for high-degree polynomial-time solvable problems (e.g., the *weighted matching problem* [6] and the *watchman route problem* [11]).

[1] Notice that for $\beta = 1/2$, all the graph edges have the same cost, and the problem becomes trivial.

[2] This bound can be obtained after conditioning the costs of the input graph G in such a way that the problem of determining a minimum-cost cycle cover of G, which represents the core procedure of the algorithm in [2], reduces to that of finding a maximum-cost simple 2-matching in the transformed graph, which can be solved in $O(n^3)$ time [7].

Actually, designing a linear-time approximation algorithm for the β-MTSP starting from the efficient implementation given in [8] of the Christofides algorithm sounds prohibitive, since this uses the very efficient $O(n^{2.5}\log^{1.5} n)$ time approximate minimum-cost perfect matching algorithm as a brick for constructing a feasible solution (and in the worst case such an algorithm runs on a graph having $\Theta(n)$ vertices). The same holds for the algorithm of Böckenhauer *et al.* [2], which makes use of the long-standing $O(n^3)$ time procedure for computing a maximum-cost perfect 2-matching. Thus, a different approach needs to be used.

A well-known approximation algorithm for the metric TSP is the *double-tree shortcutting* algorithm, which works as follows: first, construct a *Minimum Spanning Tree* (MST, for short) T of G; after, construct a Eulerian tour D on the multigraph $G' = T \cup T$, and finally return a Hamiltonian cycle H from D by shortcutting in G repeated vertices in the Eulerian tour. This algorithm (which we will call `Double-MST shortcut` in the rest of the paper) is easily seen to guarantee a 2-approximation ratio, and then it turns out to be a $\frac{2\beta^2}{2\beta^2-2\beta+1}$-approximation algorithm for the β-MTSP [2].[3] As a matter of fact, it is then always outperformed by its super-linear counterparts. In an effort of improving this gap, by maintaining the linear-time constraint, we therefore develop an easy modification of such an algorithm, which allows us to obtain, by means of an accurate analysis, a 2β-approximate solution, thus beating the `Double-MST shortcut` for any $1/2 < \beta < 1$. This improvement is effective, since as a side result, we show that for any $1/2 < \beta < 1$, the performance ratio of the `Double-MST shortcut` is strictly larger than 2β. As a matter of fact, the reduction of the theoretical gap with respect to the superlinear approximation algorithm is significant: for instance, for all $1/2 < \beta \leq 3/4$, our algorithm is only about 5% away from it in the worst case, while for the `Double-MST shortcut` this gap raises to about 26%.

It is worth noticing that the `Double-MST shortcut` approach is at the basis of one of the best-performing heuristics available for the TSP, namely that developed by Deineko and Tiskin[3,5]. This heuristic computes in $O(4^d n^2)$ time (where d denotes the maximum node-degree in T) the minimum cost Hamiltonian cycle that can be obtained using the `Double-MST shortcut` approach. From now on, we will call this heuristic with `Double-MST Min-weight shortcut`. Extensive computational experiments have shown that this strategy allows very good approximations, often better that those obtained with heuristics derived by the Christofides algorithm, although it requires a running time which might be exponential. However, despite its high performances registered in practice, no theoretical approximation guarantee better than that of `Double-MST shortcut` was exhibited in [5]. Our analysis immediately shows that the algorithm given in [5] has actually an approximation ratio of 2β, for any $\beta < 1$. Moreover, we also prove that this ratio is asymptotically tight. Due to the practical relevance

[3] This ratio results from the fact that if $\mathcal{A}$ is an α-approximation algorithm for the metric TSP, then $\mathcal{A}$ is an $\frac{\alpha\cdot\beta^2}{\beta^2+(\alpha-1)(1-\beta)^2}$ -approximation algorithm for the β-MTSP (see [2] for further details).

of heuristics based on the minimum-weight shortcutting of D, we see this as a noteworthy consequence of our results.

The rest of the paper is organized as follows: in Section 2 we show that the `Double-MST shortcut` algorithm does not return a 2β-approximate solution; in Section 3 we present an enhanced version of the previous algorithm and prove it computes a 2β-approximate solution; in Section 4 we prove that `Double-MST Min-weight shortcut` does not compute a solution within a constant factor smaller than 2β, thus proving that no algorithm based on the `Double-MST shortcut` approach can be significantly better than the one presented in the previous section. Finally, in Section 5, we provide a graphical comparison (in terms of approximation ratio) between our algorithm and its counterparts.

2 `Double-MST Shortcut` Is Not a 2β-Apx Algorithm

From now on, we will assume that the input graph $G = (V, E)$ is an instance of the β-MTSP. Moreover, paths and cycles will be expressed explicitly throughout the sequence of constituting vertices. The `Double-MST shortcut` algorithm does not prescribe how to shortcut repeated vertices in the Eulerian cycle obtained by merging the two copies of the MST. An easy to state and widely used rule, is to perform shortcut according to the visiting order of vertices in a *depth-first search* of the MST. Applying this rule, the `Double-MST shortcut` algorithm can be stated as follows:

Algorithm 1. An implementation of the `Double-MST shortcut` algorithm

Input: A β-metric graph $G = (V, E)$.
Step 1. Construct an MST T of G.
Step 2. Perform a depth-first search on T starting from an arbitrary vertex v_0. Let $v_0, v_1, \ldots, v_{n-1}$ be the sequence of vertices in the order they are visited.
Output: Return the cycle $H = v_0 v_1 \ldots v_{n-1} v_0$ of G.

As said in the introduction, this algorithm turns out to be a $\frac{2\beta^2}{2\beta^2-2\beta+1}$-approximation algorithm for the β-MTSP. Although this ratio has not been proven to be tight, in the following we provide an input instance showing that the cost of the Hamiltonian cycle computed by `Double-MST shortcut` can be strictly larger than 2β times the optimal one.

For a given $\frac{1}{2} < \beta < 1$, let $G = (V, E)$ be a β-metric graph with vertex set $V = \{v_0, x_1, y_1, z_1, \ldots, x_h, y_h, z_h\}$, and such that the edge costs are defined as follows:

- $c(v_0, x_i) = 1, c(v_0, y_i) = \frac{\beta}{1-\beta}, c(v_0, z_i) = 2\beta, \forall 1 \leq i \leq h$;
- $c(y_i, v) = \frac{\beta}{1-\beta}$, $\forall 1 \leq i \leq h$ and $\forall v \in V \setminus \{y_i\}$;
- $c(x_i, x_j) = 2\beta$, $\forall 1 \leq i, j \leq h$;
- $c(z_i, z_j) = 2\beta$, $\forall 1 \leq i, j \leq h$;

– $c(x_i, z_j) = 1$, $\forall 1 \le i \le j \le h$;
– $c(x_i, z_j) = 2\beta^2 + \beta$, $\forall 1 \le j < i \le h$.

An MST T of G is formed by edges $(v_0, x_i), (x_i, y_i), (x_i, z_i)$, for each $1 \le i \le h$.In Figure 1 the MST for the case $h = 3$ is depicted. Assume now that the `Double-MST shortcut` algorithm takes v_0 as root of T, and performs a depth-first search visiting y_i before z_i, for all $1 \le i \le h$. Then, the algorithm builds the solution $H = v_0x_1y_1z_1x_2y_2z_2 \dots x_hy_hz_hv_0$, having cost (see Figure 2 for the case $h = 3$)

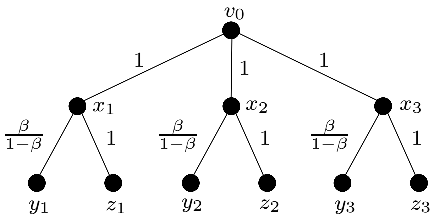

Fig. 1. An MST of G

$$c(H) = \frac{1}{1-\beta}\left(h(3\beta + \beta^2 - 2\beta^3) + 1 - 3\beta^2 + 2\beta^3\right).$$

Consider the solution $H^* = y_1y_2 \dots y_hv_0x_hz_hx_{h-1}z_{h-1} \dots x_1z_1y_1$ which has cost (see Figure 2 for the case $h = 3$)

$$c(H^*) = \frac{1}{1-\beta}\left(h(2-\beta) + \beta\right).$$

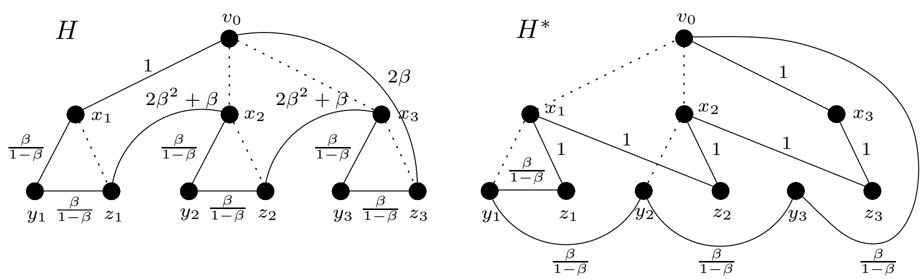

Fig. 2. The solution H constructed by the `Double-MST shortcut` algorithm on the left, and the solution H^* on the right

Therefore, if OPT denotes an optimal solution, we have

$$\frac{c(H)}{c(\text{OPT})} \ge \frac{c(H)}{c(H^*)} = \frac{h(3\beta + \beta^2 - 2\beta^3) + 1 - 3\beta^2 + 2\beta^3}{h(2-\beta) + \beta} > 2\beta,$$

where the last inequality holds whenever $h > \frac{2}{\beta(1-\beta)} - \frac{1-\beta}{\beta}$.

3 The 2β-Approximation Algorithm

In this section we provide a refinement of `Double-MST shortcut` which returns a 2β-approximate solution for the β-MTSP (its pseudo-code is given in Algorithm 2). In the following, given a vertex v of the considered rooted tree T, we denote by T_v the subgraph of T induced by v and by its children in T. We say that $v' \in T_v$ is a *closest child* of v if $c(v,v') \leq c(v,v''), \forall v'' \in T_v$.

Algorithm 2. The `Refined Double-MST shortcut` algorithm

Input: A β-metric graph $G=(V,E)$.

Step 1. Construct an MST T of G.

Step 2. Root T at any arbitrary vertex of degree 1. Starting from the root, do a *careful* depth-first search of T, i.e., a depth-first search with the following additional rule: when the visit of an internal vertex of T begins, choose one of its closest children as the next vertex to visit. Let $v_0, v_1, \ldots, v_{n-1}$ be the sequence of vertices in the order they are visited.

Output: Return the cycle $H = v_0 v_1 \ldots v_{n-1} v_0$ of G.

3.1 Algorithm Analysis

In this section we prove that `Refined Double-MST shortcut` returns a 2β-approximate solution. The core idea of the proof is to use properties of a careful depth first search together with the fact that in β-metric graphs, the cost of a direct edge between two vertices u and v is never larger than β times the cost of any other path between u and v. More formally, a simple proof by induction shows that

Lemma 1. *Let $P = v_0 v_1 \ldots v_k$ be a simple path in G of length $k \geq 3$. Then*

$$c(v_0,v_k) \leq \beta\, c(P) - (1-\beta) \sum_{j=1}^{k-2} \left(\beta^j \sum_{i=0}^{k-1-j} c(v_i, v_{i+1}) \right).$$

□

Let us denote by A the set of edges of T belonging to H, and by $B = E(H) \setminus A$ the remaining edges in H. The following lemma holds:

Lemma 2. *For every non-leaf vertex v_j of T, we have that $|A \cap E(T_{v_j})| = 1$.*

Proof. Clearly, the lemma holds for $j=0$. Hence, assume that $j>0$. There are exactly two edges in $E(H)$ incident to v_j. One of them connects v_j to v_{j-1}, so it does not belong to T_{v_j} (indeed, for the depth-first search properties, v_{j-1} cannot be a descendent of v_j). If v_j is not a leaf, one of its children is visited immediately after v_j, hence $(v_j, v_{j+1}) \in E(T_{v_j})$. □

Observe that edges in A are exactly edges of the form (u,v), where u is an internal node of T while v is a closest child of u. Moreover, observe that since the root v_0 has degree 1, then edges $(v_0,v_1), (v_1,v_2)$ always belong to A. For any edge $e \in E(H)$, let P_e denote the unique path in T joining the two endpoints of e (notice that if $e \in A$, then P_e consists of e). Because of the depth-first search properties, it is not hard to see that, for any $e \in E(T)$, there are exactly two edges of H, say e_1, e_2, that *cover* e, i.e., $e \in E(P_{e_1})$ and $e \in E(P_{e_2})$.

Let us denote by OPT an optimal solution and by e^* an edge of OPT of maximum cost. As OPT minus e^* is a spanning tree of G, we can claim that $c(\text{OPT}) \geq c(T) + c(e^*)$. In the remaining part of this section we will prove that $c(H) \leq 2\beta\big(c(T) + c(e^*)\big)$, thus proving that $c(H) \leq 2\beta\, c(\text{OPT})$. The idea of the proof is to charge the cost of H to the edges in T plus e^* in such a way that each of these edges will be charged with at most 2β times its cost. This can be done in a simple way because every edge f in H is associated with P_f, a path in T. Therefore, it is quite natural to start charging the cost of f to the edges of P_f according to the formula in Lemma 1. Let e be an edge of T, and let $e_1, e_2 \in E(H)$ be such that P_{e_1}, P_{e_2} cover e. Using the formula in Lemma 1, we can observe the following

- if $e \notin A$ then e is charged with at most 2β times its cost, as both P_{e_1}, P_{e_2} are different from e;
- if $e \in A$ then e is charged with at most $1 + \beta$ times its cost, as either P_{e_1} or P_{e_2} must be different from e.

Therefore, the unique problem to solve is how to uncharge edges in A and still keep every other edge with a charge of 2β times its cost. By the careful depth-first search, we know that A contains only edges of the form (u, v), where u is an internal node of T while v is a closest child of u. By definition of closest child, we have that $c(u, v) \leq c(u, v'), \forall (u, v') \in E(T_u)$. Moreover, it is possible to prove that $c(e) \leq c(e^*), \forall e \in E(T)$. Finally, in β-metric graphs, with $\frac{1}{2} \leq \beta < 1$, two adjacent edges f, f' satisfies $c(f) \leq \frac{\beta}{1-\beta}\, c(f')$ (see [2]). Since from Lemma 1, some edges may be charged with less than 2β times their cost, we can therefore uncharge (u, v) using the quantities we are saving on its adjacent edges in T plus e^* using the above rules. In what follows, we will prove that this can always be done in such a way that each edge of T plus e^* will be charged by at most 2β times its cost. The quality of the solution computed by `Refined Double-MST shortcut` is better than the one computed by `Double-MST shortcut` because, whenever an edge $(u, v') \in E(T_u)$ has been charged with $(2\beta - \delta)c(u, v')$, thanks to the careful depth-first search, we can uncharge (u, v) by $\delta\, c(u, v)$ instead of uncharging it by the smaller value $\frac{1-\beta}{\beta}\, \delta\, c(u, v)$. Before providing a formal proof of the above idea, we point out that the restriction of having a vertex of degree 1 as the root of T is only for the purpose of avoiding technicalities. All the results contained in this section can be extended to the general case in which T is rooted at any arbitrary vertex.

For any $e \in E(H)$, with $P_e = x_0 x_1 \ldots x_k$, we define V_e as the set of vertices $\{x_1, x_2, \ldots, x_{k-2}\}$, and $V' = \bigcup_{e \in E(H), |E(P_e)| \geq 3} V_e$. The following holds:

Lemma 3. *Let $e = (v_j, v_{j+1}) \in E(H)$ be such that $P_e = x_0 x_1 \ldots x_k$, with $x_0 = v_j$, $x_k = v_{j+1}$, and assume that $k \geq 3$. Then, for $1 \leq i \leq k - 1$, x_i is the parent in T of x_{i-1}.*

Proof. Since P_e is a simple path in T, all the vertices in P_e but one, say x_h, have their parent in T contained in P_e. Suppose by contradiction that $h < k-1$. Then x_k is a descendent of x_{h+1} in T. It follows that x_{h+1} is visited after x_0 and before x_k. But this is not possible since $(x_0, x_k) \in E(H)$ implies that x_k is visited immediately after x_0. Therefore, it must be either $h = k-1$ or $h = k$. In both cases, this implies that x_{k-1} is the parent of x_{k-2} in T, which in its turn is the parent of x_{k-3} in T, and so on. From this, the claim follows. □

Let $\mathcal{L}$ denote the set of leaves of T. We have that

Lemma 4. $V' = V \setminus (\mathcal{L} \cup \{v_0, v_1\})$.

Proof. Lemma 3 implies that V' cannot contain any of the vertices in $\mathcal{L} \cup \{v_0, v_1\}$. Hence, in order to prove the claim, it is enough to show that any internal vertex v_j of T but v_0 and v_1 belongs to V_e, for some $e \in E(H)$. As v_j is an internal vertex and because of the depth-first search properties, there is an edge in H, say $e = (v_{h-1}, v_{h \bmod n})$, with $h-1 > j$, such that v_{h-1} is a proper descendent of v_j while $v_{h \bmod n}$ is not a descendent of v_j. Then, in order to conclude that $v_j \in V_e$, it suffices to observe that from the depth-first search properties, $v_{h \bmod n}$ cannot be the parent of v_j in T. □

From now on, we will denote by A' the set of edges in A but $(v_0, v_1), (v_1, v_2)$. Moreover, we define a function $\sigma : A' \to V'$ which maps each edge $e \in A'$ into the endpoint of e which is the parent in T of the other endpoint. From the above lemma and from Lemma 2, it is easy to derive that σ is a bijective function. We can now prove the following

Lemma 5. *For every $e \in B$ such that $P_e = x_0 x_1 \ldots x_k$ contains $k \geq 3$ edges, we have that*

$$\sum_{j=1}^{k-2} \left(\beta^j \sum_{i=0}^{k-1-j} c(x_i, x_{i+1}) \right) \geq \sum_{x \in V_e} c(\sigma^{-1}(x)). \tag{1}$$

Proof. Grouping together terms with respect to the edge costs, the left-hand side of (1) becomes

$$\sum_{j=1}^{k-2} \beta^j c(x_0, x_1) + \sum_{i=1}^{k-2} \sum_{j=1}^{k-1-i} \beta^j c(x_i, x_{i+1}). \tag{2}$$

By suitably rearranging its terms, (2) can be rewritten as

$$\sum_{i=1}^{k-2} \left(\sum_{j=1}^{k-1-i} \beta^j c(x_{i-1}, x_i) + \beta^{k-1-i} c(x_i, x_{i+1}) \right). \tag{3}$$

Now we bound the i-th term of the external summation in (3) with respect to the cost of edge $\sigma^{-1}(x_i)$. Recall that (see [2]) for any two adjacent edges e_1, e_2 of G, it is $c(e_1) \leq \frac{\beta}{1-\beta} c(e_2)$. Then, for each $x_i \in V_e$ it is $c(\sigma^{-1}(x_i)) \leq$

$\frac{\beta}{1-\beta}\,c(x_i,x_{i+1})$. Moreover, $c(\sigma^{-1}(x_i)) \le c(x_{i-1},x_i)$, since $\sigma^{-1}(x_i)$ and (x_{i-1},x_i) are both edges of T_{x_i}. Then we have

$$\begin{aligned}
(3) &\ge \sum_{i=1}^{k-2}\left(\sum_{j=1}^{k-1-i}\beta^j c(\sigma^{-1}(x_i)) + \beta^{k-2-i}(1-\beta)\,c(\sigma^{-1}(x_i))\right)\\
&= \sum_{i=1}^{k-2}\left(\sum_{j=1}^{k-2-i}\beta^j + \beta^{k-2-i}\right)c(\sigma^{-1}(x_i))\\
&\ge \sum_{i=1}^{k-2}\left(\sum_{j=1}^{k-2-i}\frac{1}{2^j} + \frac{1}{2^{k-2-i}}\right)c(\sigma^{-1}(x_i)) = \sum_{i=1}^{k-2} c(\sigma^{-1}(x_i)) = \sum_{x\in V_e} c(\sigma^{-1}(x))
\end{aligned}$$

where the last inequality holds because $\beta \ge \frac{1}{2}$. □

We are now ready to give our main result:

Theorem 1. *The* `Refined Double-MST shortcut` *algorithm is a 2β-approximation algorithm for the β-MTSP.*

Proof. Let us start by setting $B = \bigcup_{2\le i\le n} B_i$, where $B_i = \{e \in B \text{ s.t. } |E(P_e)| = i\}$. For any $e \in E(H)$, let $P_e = x_0^e x_1^e \ldots x_k^e$. We can bound the total cost of H with

$$\begin{aligned}
c(H) &= \sum_{e\in A} c(e) + \sum_{e\in B_2} c(e) + \sum_{k=3}^{n}\sum_{e\in B_k} c(e) \le \sum_{e\in A} c(e) + \beta\sum_{e\in B_2} c(P_e) + \\
&\quad + \sum_{k=3}^{n}\sum_{e\in B_k}\left(\beta\, c(P_e) - (1-\beta)\sum_{j=1}^{k-2}\left(\beta^j\sum_{i=0}^{k-1-j} c(x_i^e, x_{i+1}^e)\right)\right)\\
&\le \sum_{e\in A} c(e) + \beta\sum_{k=2}^{n}\sum_{e\in B_k} c(P_e) - (1-\beta)\sum_{k=3}^{n}\sum_{e\in B_k}\sum_{x\in V_e} c(\sigma^{-1}(x))
\end{aligned}$$

where the first inequality holds from the β-metricity of G and from Lemma 1, while the last inequality holds from Lemma 5. Since each edge of T is covered by exactly two edges of H, then

$$\sum_{k=2}^{n}\sum_{e\in B_k} c(P_e) = \sum_{e\in A} c(e) + 2\sum_{e\in E(T)\setminus A} c(e).$$

By definition, observe that each $e \in E(H)$ for which $V_e \subseteq V'$ is in B_j for some $j \ge 3$. From this fact and because σ is bijective we can derive

$$\sum_{k=3}^{n}\sum_{e\in B_k}\sum_{x\in V_e} c(\sigma^{-1}(x)) \ge \sum_{x\in V'} c(\sigma^{-1}(x)) = \sum_{f\in A'} c(f)$$

(in fact, here we could prove that equality holds, since $e \ne e' \Rightarrow V_e \cap V_{e'} = \emptyset$).

To conclude the proof, let e^* be a maximum-cost edge of any optimal solution Opt. Observe that, $c(e) \leq c(e^*)$ for every $e \in E(T)$. Indeed, let us consider the cut induced by the removal of e from T, and let $e' \in E(H)$ be an edge traversing that cut. Then, since T is an MST, we have $c(e) \leq c(e') \leq c(e^*)$. Using previous observations we have

$$
\begin{aligned}
c(H) \leq\ & c(v_0, v_1) + c(v_1, v_2) + \sum_{e \in A'} c(e) \\
& + \beta \left(c(v_0, v_1) + c(v_1, v_2) + \sum_{e \in A'} c(e) + 2 \sum_{e \in E(T) \setminus A} c(e) \right) - (1 - \beta) \sum_{e \in A'} c(e) \\
& = 2\beta \left(\sum_{e \in A'} c(e) + \sum_{e \in E(T) \setminus A} c(e) \right) + (1 + \beta) \Big(c(v_0, v_1) + c(v_1, v_2) \Big) \\
& \leq 2\beta \left(\sum_{e \in A'} c(e) + \sum_{e \in E(T) \setminus A} c(e) \right) + c(v_0, v_1) + c(v_1, v_2) + 2\beta\, c(e^*) \\
& \leq 2\beta\, c(T) + 2\beta\, c(e^*) \leq 2\beta\, c(\textsc{Opt})
\end{aligned}
$$

where the last but two inequality holds because $c(e^*) \geq c(e),\ \forall e \in E(T)$, the last but one inequality holds because $\beta \geq \frac{1}{2}$, and the last inequality follows from the fact that by removing e^* from Opt, one obtains a spanning tree. □

4 Lower Bound for `Double-MST Min-Weight Shortcut`

In the introduction we have mentioned one of the best heuristics available for the TSP, i.e., the heuristic we called `Double-MST Min-weight shortcut` [3,5]. We said that this heuristic computes in $O(4^d n^2)$ time (where d denotes the maximum node-degree in T) the minimum cost Hamiltonian cycle that can be obtained using the `Double-MST shortcut` approach. It is worth noticing that the result proved for the algorithm we proposed in Section 3 immediately implies that `Double-MST Min-weight shortcut` is a 2β-approximation algorithm. In this section we prove that `Double-MST Min-weight shortcut` cannot compute an approximate solution within a factor which is significantly better than 2β. As a consequence, 2β is an asymptotic lower bound for the approximation ratio of all the algorithms based on the `Double-MST shortcut` approach. Hence, the `Refined Double-MST shortcut` is one of the best approximation algorithms based on this approach.

Consider the β-metric graph of $n+1$ vertices given in Figure 3. All the internal vertices of the MST T represented with solid edges have degree $\sqrt{n}$. A feasible solution is given by the $\sqrt{n}$ dotted paths arbitrarily linked one another and with x to form a Hamiltonian cycle. As the cost of each dotted path is $\sqrt{n}-1$ and as the cost of any other edge is at most 2β, we have that the cost of an optimal solution Opt is upper bounded by $c(\textsc{Opt}) \leq \sqrt{n}(\sqrt{n} - 1) + 2\beta(\sqrt{n} + 1) \leq n + 2\beta\sqrt{n}$.

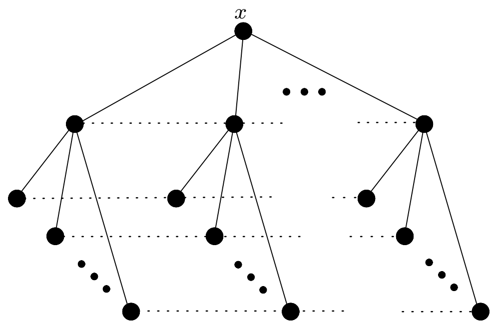

Fig. 3. A β-metric graph of $n+1$ vertices showing the asymptotic lower bound of 2β on the approximation ratio of `Double-MST Min-weight shortcut`. Both solid and dotted edges have cost 1, while the cost of missing edges is 2β. An MST T is given by the set of solid edges The degree of all the internal vertices of the MST is $\sqrt{n}$.

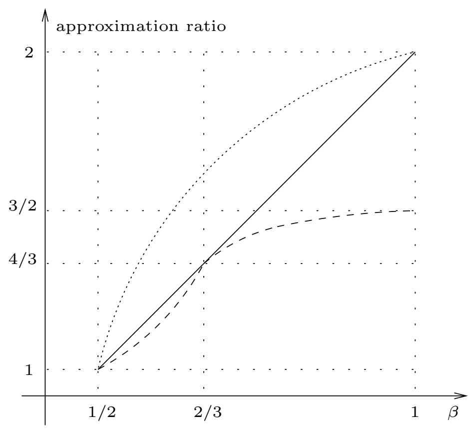

Fig. 4. The approximation ratio of our algorithm (solid line), as compared to that provided by the `Double-MST shortcut` (dotted) and the currently best-known approximation algorithm (dashed)

Concerning the solution built by `Double-MST Min-weight shortcut`, first of all notice that no Hamiltonian cycle contains more than $2\sqrt{n}$ solid edges, as these edges are all incident to the $\sqrt{n}$ internal vertices of T but x, and the degree of every vertex in a Hamiltonian cycle is 2. Now, observe that every dotted edge is a shortcut of some path in $D = T \cup T$ containing 2 solid edges which are incident to x. As the degree of x in D is $2\sqrt{n}$, we have that every solution built by `Double-MST Min-weight shortcut` contains at most $\sqrt{n}$ dotted edges. As a consequence, the solution computed by `Double-MST Min-weight shortcut` contains at least $n+1-3\sqrt{n}$ edges of cost 2β. As $c(\text{OPT}) \leq n + 2\beta\sqrt{n}$, we therefore have that `Double-MST Min-weight shortcut` does not return a $(2\beta-\epsilon)$-approximate solution, for every $\epsilon > 0$.

5 Conclusion

In Figure 4, we provide a comparison between the approximation algorithms for the β-MTSP discussed in this paper, namely the `Double-MST shortcut`, that obtained by composing the algorithms given in [4,2], and finally our one. It is worth noticing that our algorithm induces a significant improvement in the gap with respect to the superlinear approximation algorithm: for instance, for all $1/2 < \beta \leq 3/4$, our algorithm is only about 5% away from it in the worst case, while for the `Double-MST shortcut` this gap raises to about 26%. Thus, especially for this range of values of β, from a practical point of view one can make use of our very simple linear-time algorithm—instead of pursuing a very complicate implementation of the superlinear algorithm—by only paying a little bit more in terms of approximation ratio.

References

1. Arora, S.: Polynomial time approximation schemes for Euclidean traveling salesman and other geometric problems. J. ACM 45(5), 753–782 (1998)
2. Böckenhauer, H.-J., Hromkovič, J., Klasing, R., Seibert, S., Unger, W.: Approximation algorithms for the TSP with sharpened triangle inequality. Information Processing Letters 75, 133–138 (2000)
3. Burkard, R.E., Deineko, V.G., Woeginger, G.J.: The travelling salesman and the pq-tree. Mathematics of Operations Research 23(3), 613–623 (1998)
4. Christofides, N.: Worst-case analysis of a new heuristic for the traveling salesman problem. Technical report, Graduate School of Industrial Administration, Carnegy–Mellon University (1976)
5. Deineko, V.G., Tiskin, A.: Fast minimum-weight double-tree shortcutting for metric TSP. In: Demetrescu, C. (ed.) WEA 2007. LNCS, vol. 4525, pp. 136–149. Springer, Heidelberg (2007)
6. Drake Vinkemeier, D.E., Hougardy, S.: A linear-time approximation algorithm for weighted matchings in graphs. ACM Transactions on Algorithms 1(1), 107–122 (2005)
7. Gabow, H.N.: An efficient reduction technique for degree-constrained subgraph and bidirected network flow problems. In: Proc. of the 15th Annual ACM Symp. on Theory of Computing (STOC), pp. 448–456 (1983)
8. Gabow, H.N., Tarjan, R.E.: Faster scaling algorithms for general graph-matching problems. J. ACM 38(4), 815–853 (1991)
9. Papadimitriou, C.H., Vempala, S.: On the approximability of the traveling salesman problem. Combinatorica 26(1), 101–120 (2006)
10. Reinelt, G.: TSPLIB library, University of Heidelberg, `http://www.iwr.uni-heidelberg.de/groups/comopt/software/tsplib95/`
11. Tan, X.: Approximation algorithms for the watchman route and zookeeper's problems. Discrete Applied Mathematics 136(2-3), 363–376 (2004)